

Revision control with Subversion

Geoff Richards

geoff@laxan.com

WYLUG · 2006 July 10

Revision control

- For tracking source code, etc.
- Record history of changes to files
- Browsing history:
 - Changelogs
 - Diffs ([example](#))
- Centralized repository and working copy

Further notes:

Of course the old RCS didn't use a centralized repository, which made it harder to coordinate work among teams. It made heavy use of locking (only one checkout allowed at a time, unless you override) to deal with this. CVS is basically the same as RCS except that history is recorded centrally and the checkouts aren't locked.

Subversion & CVS

- Binary files
- Directories have history
- Copy-from recorded
- Properties
- No tags
- API
 - C, Python (×2), Ruby, Perl, Java

Further notes:

Subversion isn't (yet) quite as good as you might imagine at recording file move or rename operations. It does store the 'copy-from' information, but it doesn't distinguish a copy from a rename or move, except that in the latter case the original file is deleted in the same revision as the new one is created.

Revisions

- Make changes locally then commit
- Each commit becomes revision
- Revnums: r1, r2, ...
- No undo

Further notes:

With a FSFS repository (which is the default with modern versions of Subversion) it probably is possible to 'undo' a revision by carefully deleting the files created when it was committed, but I wouldn't recommend it. You can always recover from mistakes by pulling older versions of files out of earlier revisions.

Example

- Create repository (`svnadmin create`)
- Get working copy (`svn co`)
- Add directories and files:
 - `svn mkdir`
 - `svn add`
- Manage revisions:
 - `svn status`
 - `svn ci`
 - `svn update`

Further notes:

Here's roughly what I did as an example:

```
svnadmin create repos
svn co file:///$(pwd)/repos foo
cd foo
svn mkdir trunk tags branches
svn st
svn ci
```

```
svn mkdir trunk/src
vim trunk/src/hello.pl
svn st
svn add trunk/src/hello.pl
svn ci
```

```
# edit the file
svn ci
svn st
svn diff
```

```
svn log
svn up
svn log
svn log -v --xml
svn log -v
```

```
svn diff -r 2:3
```

Branches & tags

- Tag: bookmark point in history
- Branch: parallel streams of development
- Done with copying (efficiently)
- Leave places in repository layout:
 - trunk
 - tags
 - branches
- Branches and tags can be safely deleted

Further notes:

These three top-level directories are just normal directories. There's nothing special about them, and their names and locations are just a convention. You can organize repositories differently if you want.

So to create a tag or branch you just copy the 'trunk' directory from the revision you want to start with. You can make changes in a branch in parallel with the mainline development, and then merge them into the trunk when you're ready, although you have to keep track of which branches have been merged, and when, yourself. This is usually done by leaving annotations in log messages.

Network access

- SSH
 - `svn co svn+ssh://hostname/path foo`
- svnservice
- HTTP (WebDAV)

Further notes:

Both svnservice and HTTP access allow users to checkout a working copy without having to have an account (if that's what you want), and can be configured to use various authentication systems. SSH access is easy to set up (you just need an SSH server running on the machine with the repository) but you have to create an account for each user.

Other clients

- GUIs
 - [RapidSVN](#): cross-platform GUI client
 - [KDESVN \(history browser screenshot\)](#)
 - [TortoiseSVN](#): Windows shell extension
- SVK: Perl-based client for distributed revision control

Further notes:

TortoiseSVN is apparently stable and widely used, but the cross-platform and Linux GUI clients are still quite young.